RSAC 2025 Conference

Many Voices.
One Community.

SESSION ID: TPV-R03

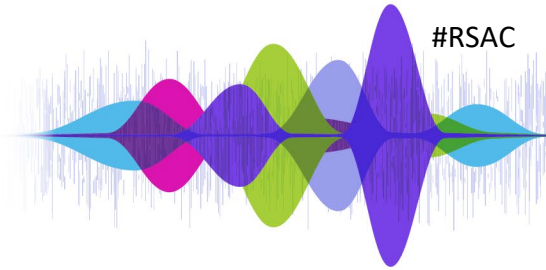
Fortifying the AI Supply Chain: Ensuring Trust from Data to Deployment

Dr. Ulrich Lang

Founder & CEO
ObjectSecurity LLC
linkedin.com/in/ulrichlang

Jason Kramer

Senior Research Software Engineer
ObjectSecurity LLC
linkedin.com/in/jasonkramerr



Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

The ObjectSecurity logo must not be removed from any slides, and any publications must include the ObjectSecurity logo in full.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

About Us

Dr. Ulrich Lang (Speaker)

- 20+ years in cybersecurity
- AI/ML for binary vuln analysis, OT/ICS security, 5G, supply chain, policy automation

Jason Kramer (Speaker)

- R&D Lead, Trusted AI/ML Cybersecurity
- AI/ML for OT/ICS security, component detection, supply chain risk, data cleansing

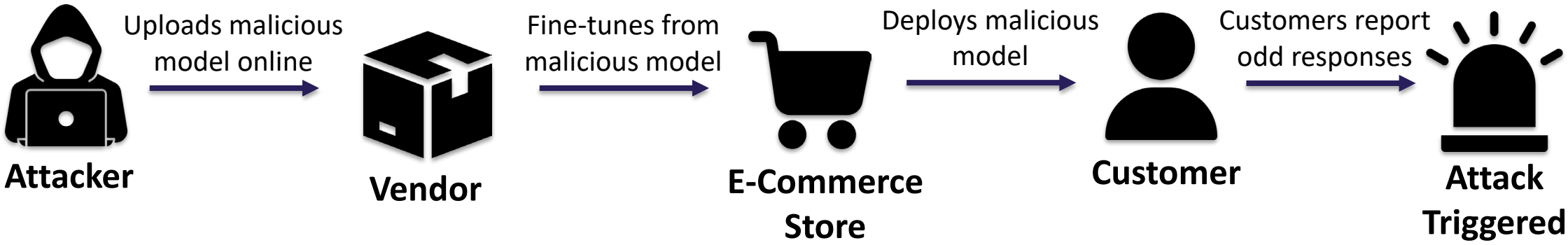
Stephen Brennan (Contributor)

- BSc Math-CS, UC San Diego
- AI/ML for AI model vulnerabilities, explainability, and mitigation

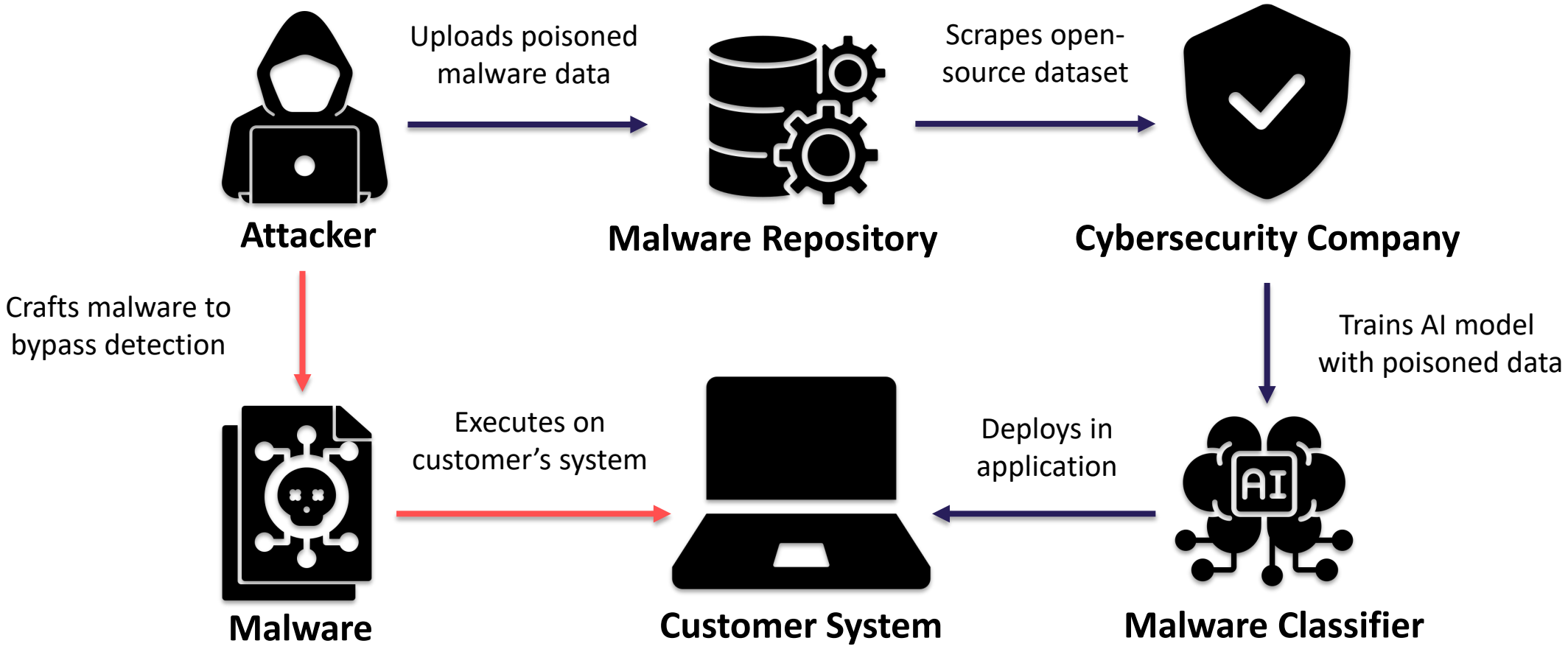
The AI Supply Chain is at Risk

If attackers can modify even one part of an AI system, how can we trust its outputs?

- AI supply chains rely on trust between vendors and users
- Security vulnerabilities exist at every step



AI Supply Chain – Data Poisoning



Why Secure the AI Supply Chain?

- Protect business operations & reliability
- Prevent litigation
- Avoid reputational damage
- Differentiate & sell more
- Improve safety

RSAC 2025 Conference

The Supply Chain is Only as Strong as Its Weakest Link

Users and vendors must improve AI supply chain security together - but how?

Many Voices.
One Community.

Challenges to Securing the AI Supply Chain

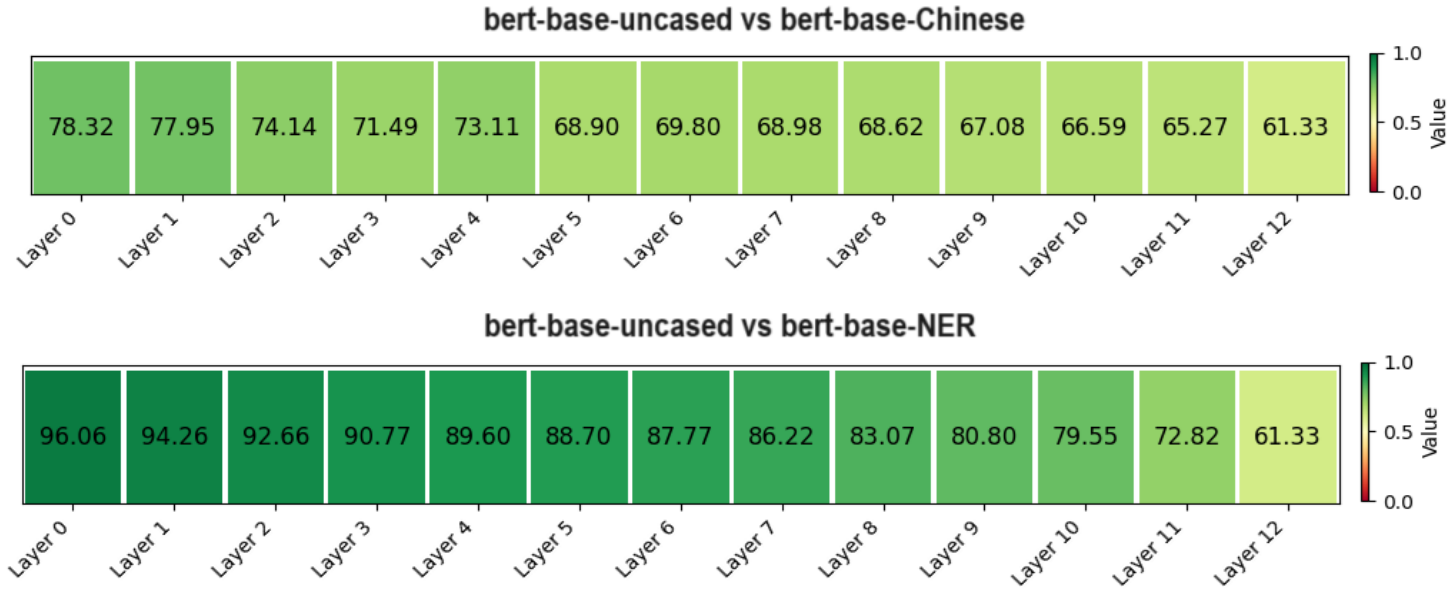
- AI isn't just code
- AI security ≠ traditional cybersecurity
- Models are often opaque boxes
- Models change behavior unpredictably
- Lack of tools to patch models when an issue arises

Supply Chain Risk – Poisoning and Evasion Attacks

- **Challenge:** Poisoned data can add risks that users don't see
- **Vendor's Role:** Verify data provenance & secure datasets
- **User's Role:** Report unexpected biases, inconsistencies

Supply Chain Risk - Transfer Learning

- **Challenge:** Fine-tuned models inherit security flaws from base models
- **Vendor's Role:** Test for inherited vulnerabilities before fine-tuning
- **User's Role:** Request security validation for AI before use



Supply Chain Risk - Malicious Dependencies

- **Challenge:** AI dependencies like models from third-party repositories can be compromised
  - Malware hidden in AI models
  - Exploits in AI libraries
- **Vendor's Role:** Verify model sources, avoid blindly trusting community uploads, use binary analysis, AI vulnerability analysis
- **User's Role:** Request AI and software security analysis results for AI supply chain transparency

Supply Chain Risk: Reverse Engineering & Model Tampering

- **Challenge:** Attackers can reverse-engineer deployed models
  - Decompile and analyze model
  - Modify weights (e.g., insert backdoors)
  - Swap in compromised models
- **Vendor's role:** Ensure model integrity and use model versioning
- **User's role:** Detect AI use, verify models

```
0x401220 (main) → 0x401080 (fopen)
0x401080 (fopen) → 0x367f630 (fopen64)
0x401220 (main) → 0x401030 (puts)
0x401030 (puts) → 0x3680e50 (_IO_puts)
0x401220 (main) → 0x401040 (fread)
0x401040 (fread) → 0x367fb30 (_IO_fread)
0x401220 (main) → 0x401310 (resnet18_v1_7)
0x401310 (resnet18_v1_7) → 0x401630 (libjit_transpose_f_0_specialized)
0x401310 (resnet18_v1_7) → 0x401850 (libjit_conv2d_f_1_specialized)
0x401310 (resnet18_v1_7) → 0x401f70 (libjit_max_pool_f_2_specialized)
0x401310 (resnet18_v1_7) → 0x402200 (libjit_conv2d_f_3_specialized)
0x401310 (resnet18_v1_7) → 0x4029c0 (libjit_convDKKC8_f_4_specialized)
0x401310 (resnet18_v1_7) → 0x403100 (libjit_stacked_kernel_5_specialized)
0x401310 (resnet18_v1_7) → 0x403210 (libjit_stacked_kernel.1_6_specialized)
0x401310 (resnet18_v1_7) → 0x403320 (libjit_convDKKC8_f_7_specialized)
0x401310 (resnet18_v1_7) → 0x403850 (libjit_conv2d_f_8_specialized)
0x401310 (resnet18_v1_7) → 0x403eb0 (libjit_convDKKC8_f_9_specialized)
0x401310 (resnet18_v1_7) → 0x4044e0 (libjit_stacked_kernel.2_10_specialized)
0x401310 (resnet18_v1_7) → 0x4045f0 (libjit_conv2d_f_11_specialized)
0x401310 (resnet18_v1_7) → 0x404c90 (libjit_stacked_kernel.3_12_specialized)
0x401310 (resnet18_v1_7) → 0x404da0 (libjit_convDKKC8_f_13_specialized)
0x401310 (resnet18_v1_7) → 0x405450 (libjit_conv2d_f_14_specialized)
0x401310 (resnet18_v1_7) → 0x405bd0 (libjit_convDKKC8_f_15_specialized)
0x401310 (resnet18_v1_7) → 0x4063a0 (libjit_stacked_kernel.4_16_specialized)
0x401310 (resnet18_v1_7) → 0x4064b0 (libjit_conv2d_f_17_specialized)
0x401310 (resnet18_v1_7) → 0x406c60 (libjit_stacked_kernel.5_18_specialized)
0x401310 (resnet18_v1_7) → 0x406d70 (libjit_convDKKC8_f_19_specialized)
0x401310 (resnet18_v1_7) → 0x409300 (libjit_conv2d_f_20_specialized)
0x401310 (resnet18_v1_7) → 0x40b8d0 (libjit_convDKKC8_f_21_specialized)
0x401310 (resnet18_v1_7) → 0x40def0 (libjit_stacked_kernel.6_22_specialized)
0x401310 (resnet18_v1_7) → 0x40e000 (libjit_conv2d_f_23_specialized)
0x401310 (resnet18_v1_7) → 0x410650 (libjit_stacked_kernel.7_24_specialized)
0x401310 (resnet18_v1_7) → 0x410760 (libjit_avg_pool_f_25_specialized)
0x401310 (resnet18_v1_7) → 0x410b60 (libjit_matmul_f_26_specialized)
0x410b60 (libjit_matmul_f_26_specialized) → 0x401070 (memset)
0x401070 (memset) → 0x36a96d0 (memset)
0x401310 (resnet18_v1_7) → 0x411190 (libjit_batchedadd_f_27_specialized)
0x401220 (main) → 0x401060 (printf)
0x401060 (printf) → 0x36606f0 (printf)
0x401220 (main) → 0x401050 (fclose)
```

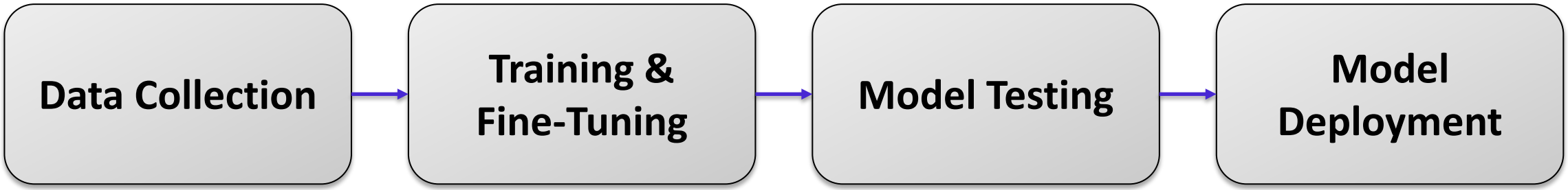
RSAC 2025 Conference

Let's Secure the AI Supply Chain Together

Key security challenges and practical solutions

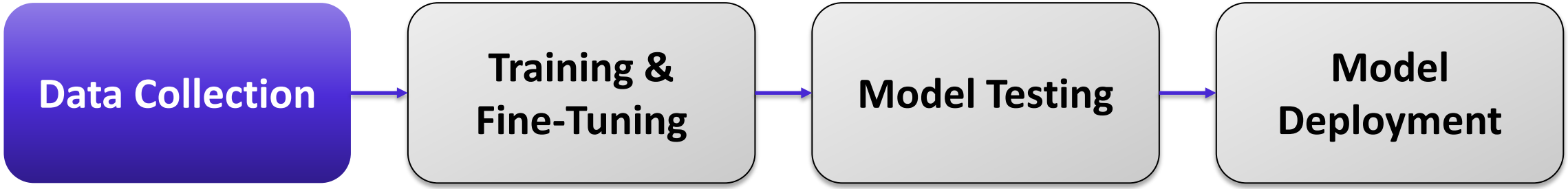
Many Voices.
One Community.

Mitigating the Supply Chain



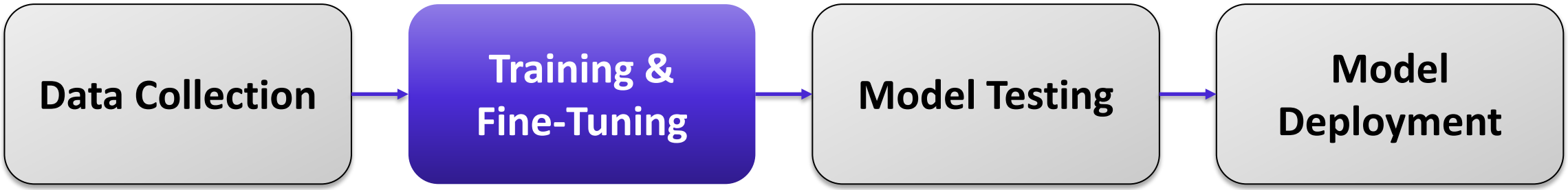
- Every step has potential vulnerabilities
- Vendors and users can work together at every step to secure the supply chain

Mitigating the Supply Chain – Data Collection



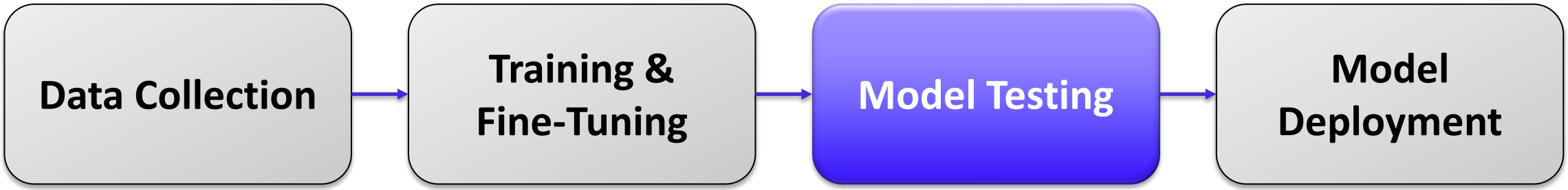
- Use data integrity verification methods
  - cryptographic hashes
  - signed datasets
- Implement data versioning tools like Data Version Control
- Manually inspect and test sample data before training
- Perform vulnerability analysis of models early in the process

Mitigating the Supply Chain – Training & Fine-Tuning



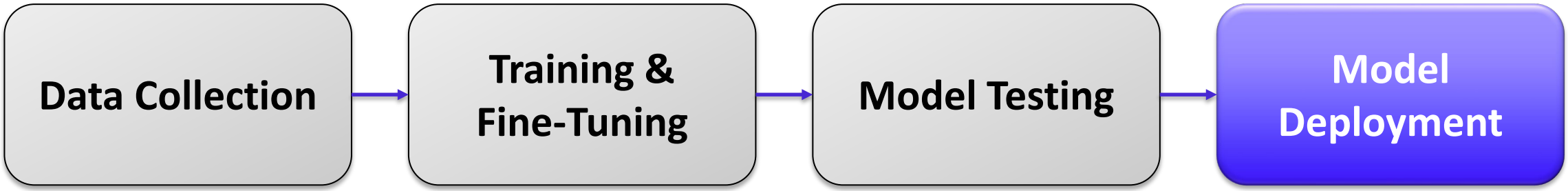
- Verify the model hash and check against malware databases
- Train with adversarial robustness techniques
- Perform vulnerability analysis of models before fine-tuning or deploying
- Use isolated and secure training environments
- Regularly track and scan dependencies for vulnerabilities

Mitigating the Supply Chain – Model Testing



- Make security documentation a requirement
- Enforce dataset provenance tracking
- Automate analysis and documentation generation for consistency and to avoid human error
- Store all training and robustness tests for reproducibility

Mitigating the Supply Chain – Model Deployment



- Continuously monitor models and adversarial test results
  - Analyze AI model drifting toward biased and unexpected outputs
- Notify affected users of any confirmed issues
- Assign a security team to handle AI reports
- Make reporting issues easy

RSAC 2025 Conference

The Future of the AI Supply Chain

Towards a more trusted and resilient AI ecosystem

Many Voices.
One Community.

Vendor Takeaways

- Use security and transparency as selling points
- Verify data sources and training provenance
- Track dependencies and run regular security checks
- Give users tools to monitor and report issues
- Patch models dynamically when vulnerabilities arise

User Takeaways

- Don't dismiss issues as false positives
- Beware of a false sense of security
- Demand security testing of models, security report
- Test models when receiving them, report vulnerabilities to vendors
- Monitor AI post-deployment for unexpected behaviors

Industry Takeaways

- Regulations are coming!
- Need for industry-wide reporting of vulnerabilities
- Settle on a standardized and enforced AIBOM format
- Define the minimum requirement of AI security assessments

Apply What You Have Learned

- **Prioritize security** throughout the AI lifecycle — from development to deployment
- **Test and monitor AI systems** continuously — assume they're dynamic
- **Collaborate and communicate** across vendors, users, and regulators
- **Avoid a false sense of security** — perform your own security analysis
- **Stay proactive** — regulations and threats are evolving fast

RSAC™ 2025 Conference

Many Voices.
One Community.

Any questions?

Image References

- Hacker: Talha Dogar (flaticon.com/free-icon/hacker_3518775)
- Box: Pixel perfect (flaticon.com/free-icon/box_726546)
- Cart: Maxim Basinski Premium (flaticon.com/free-icon/cart_665865)
- User: Freepik (flaticon.com/free-icon/user_456212)
- Alarm: Freepik (flaticon.com/free-icon/alert_2780146)
- Server Control: bsd (flaticon.com/free-icon/server-control_11192986)
- Protection: Kiranshanstry (flaticon.com/free-icon/shield_709701)
- Virus Attack: MindWorlds (flaticon.com/free-icon/virus-attack_10057679)
- Laptop Computer: Liyuza (flaticon.com/free-icon/laptop-computer_9318633)
- AI Cloud: Vectorslab (flaticon.com/free-icon/ai-cloud_16986895)